

CLAIMS

What is claimed is:

1 1. A method for processing a database query on a set of data in a database management
2 system having a data manager and an index manager, the method comprising the steps of:

3 a) in response to a data manager call to locate a data identifier in an index
4 corresponding to a selected key value, performing the step of:

5 i) locating the data identifier in the index for the selected key value; and
6 b) continuing to carry out the index-data fetch for another data identifier if there
7 is another data identifier for the selected key value in the index and the index manager
8 receives a specific condition from the data manager.

1 2. The method of claim 1 wherein step a) further comprises:

2 ii) determining whether the data specified by the data identifier is to be
3 returned to a runtime; and

4 iii) allowing the data manager to locate the data specified by the identifier
5 and make a determination.

1 3. The method of claim 2 wherein the determination involves determining whether a
2 predicate check or a data consuming operation on the data are possible.

1 4. The method of claim 3 wherein the predicate check or the data consuming operation

is performed if it is possible to perform the predicate check or the data consuming operation and a specific condition is returned to the index manager.

5. The method of claim 4 wherein the specific condition comprises a no data return condition.

6. The method of claim 2 wherein the index comprises a plurality of pages having index nodes and the method further comprises the steps of:

c) allowing the index manager to stabilize a page containing a node to be accessed in the index to locate the data identifier for the selected key value wherein the index manager does not release the stabilization of the page during a callback to the data manager.

7. The method of claim 1 wherein the index comprises a plurality of pages having index nodes and the database management system supports a set of access processes potentially able to access the index, the method further comprising the steps of:

c) determining if one of the set of access processes is waiting for an exclusive latch for accessing the index;

d) utilizing the index manager to copy the data identifier and key value from the index if there is such an access process; and

e) releasing stabilization of any pages in the index prior to any callback to the data manager.

1 8. A computer readable medium containing program instructions processing a database
2 query on a set of data in a database management system having a data manager and an index
3 manager, the program instructions comprising the steps of:

4 a) in response to a data manager call to locate a data identifier in an index
5 corresponding to a selected key value, performing the step of:

6 i) locating the data identifier in the index for the selected key value; and

7 b) continuing to carry out the index-data fetch for another data identifier if there
8 is another data identifier for the selected key value in the index and the index manager
9 receives a specific condition from the data manager.

1 9. The computer readable medium of claim 8 wherein step a) further comprises:

2 ii) determining whether the data specified by the data identifier is to be
3 returned to a runtime; and

4 iii) allowing the data manager to locate the data specified by the identifier
5 and make a determination.

1 10. The computer readable medium of claim 9 wherein the determination involves
2 determining whether a predicate check or a data consuming operation on the data are
3 possible.

1 11. The computer readable medium of claim 10 wherein the predicate check or the data
2 consuming operation is performed if it is possible to perform the predicate check or the data
3 consuming operation and a specific condition is returned to the index manager.

1 12. The computer readable medium of claim 11 wherein the specific condition comprises
2 the no data return condition.

1 13. The computer readable medium of claim 9 wherein the index comprises a plurality of
2 pages having index nodes and the program instructions further comprises the steps of:

3 c) allowing the index manager to stabilize a page containing a node to be
4 accessed in the index to locate the data identifier for the selected key value wherein the
5 index manager does not release the stabilization of the page during a callback to the data
6 manager.

1 14. The computer readable medium of claim 9 wherein the index comprises a plurality of
2 pages having index nodes and the database management system supports a set of access
3 processes potentially able to access the index, the program instructions further comprising
4 the step of:

5 c) determining if one of the set of access processes is waiting for an exclusive
6 latch for accessing the index;

7 d) utilizing the index manager to copy the data identifier and key value from the
8 index if there is such an access process; and

9 e) releasing stabilization of any pages in the index prior to any callback to the
10 data manager.

1 15. A method for processing a database query on a set of data in a database management

2 system, the method comprising the steps of:

3 a) performing an index-data fetch to locate an identified portion of a set of data
4 corresponding to a located data identifier wherein the index-data fetch comprises:

5 i) locating the data identifier in an index for a specified key value.

6 b) processing a database query having a potential push-down index-data fetch,
7 wherein step i) comprises the use of a callback to perform step a) for carrying out the
8 potential push-down for successive data identifiers in the index for the specified key value;

1 16. The method of claim 15 wherein a subset of the set of data is accessed without the
2 subset being returned to the user and the data management system supports an index data
3 fetch operation and a pushdown index-data fetch operation.

1 17. The method of claims 16 wherein if there is no data identifier for the specified key
2 value, step a) further comprises returning a not found indicator.

1 18. The method of claim 17 wherein the index comprises a plurality of pages having
2 index nodes and the step i) further comprises:

3 i1) stabilizing a page containing a node to be accessed in the index for locating
4 the data identifier for the selected key value wherein the stabilization of the page is released
5 only after completion of any callback index data fetches.

1 19. The method of claim 18 wherein the index comprises a plurality of pages having
2 index nodes wherein the database management system supports a set of access processes

3 capable of accessing the index, the method further comprising:

4 c) determining if one of the set of access processes is waiting for an exclusive
5 latch for accessing the index;

6 d) copying the data identifier and key value from the index if there is such an
7 access process; and

8 e) releasing stabilization of any pages in the index prior to any callback to
9 perform step a).

1 20. A computer readable medium containing program instructions for processing a
2 database query on a set of data in a database management system, the program instructions
3 comprising the steps of:

4 a) performing an index-data fetch to locate an identified portion of a set of data
5 corresponding to a located data identifier wherein the index-data fetch comprises:

6 i) locating the data identifier in an index for a specified key value;

7 b) processing a database query having a potential push-down index-data fetch,
8 wherein step i) comprises the use of a callback to perform step a) for carrying out the
9 potential push-down for successive data identifiers in the index for the specified key value.

1 21. The computer readable medium of claim 20 wherein a subset of the set of data is
2 accessed without the subset being returned to the user and the data management system
3 supports an index data fetch operation and a pushdown index-data fetch operation.

1 22. The computer readable medium of claims 21 wherein if there is no data identifier for

the specified key value, step a) further comprises returning a not found indicator.

23. The computer readable medium of claim 22 wherein the index comprises a plurality of pages having index nodes and the step i) further comprises:

i1) stabilizing a page containing a node to be accessed in the index for locating the data identifier for the selected key value wherein the stabilization of the page is released only after completion of any callback index data fetches.

24. The computer readable medium of claim 22 wherein the index comprises a plurality of pages having index nodes wherein the database management system supports a set of access processes capable of accessing the index, the method further comprising:

c) determining if one of the set of access processes is waiting for an exclusive latch for accessing the index;

d) copying the data identifier and key value from the index if there is such an access process; and

e) releasing stabilization of any pages in the index prior to any callback to perform step a).

25. A query processing system comprising:

a data manager coupled to an index manager, the data manager supporting predicate checking and data consuming operations, the data manager comprising:

means for determining if a predicate check or data consuming operation on a defined subset of a set of data is possible;

6 means for performing an index-data fetch wherein the means for performing
7 an index-data fetch include means for requesting a callback from the index manager if a
8 predicate check or data consuming operation is possible wherein the index manager
9 comprises:

10 means for initiating a callback to the data manager for an index-specified
11 subset of the set of data to determine if a predicate check or data consuming operation on the
12 index-specified subset of the set of data is possible;

13 means for continuing to carry out the index fetch for a data identifier if there
14 is a data identifier for a selected key value in the index and the index manager receives a
15 specific condition from the callback.

1 26. The query processing system of claim 25 wherein the means for performing the
2 index-data fetch further includes means for carrying out the operations and returning a no
3 data condition to the index manager if a predicate check or a data consuming operation is
4 possible.

1 27. The query processing system of claim 26 wherein the means for continuing the index
2 data fetch is invoked if the index manager receives a no data return condition from a
3 callback to the data manager and the means for continuing the index data fetch further
4 includes means for returning a result of no index entry found if there are no remaining data
5 identifiers.

1 28. The query processing system of claim 27 wherein the index comprises a plurality of

2 pages having index nodes, the query processing system further comprises:

3 means for allowing the index manager to stabilize a page containing a node to be
4 accessed in the index to locate the data identifier for the selected key value wherein the
5 index manager does not release the stabilization of the page during a callback to the data
6 manager.

1 29. The query processing system of claim 27 wherein the query processing system
2 further comprises:

3 stabilization means for determining if one of a set of access processes is waiting for
4 an exclusive latch for accessing the index, the stabilization means comprising:

5 means for copying the data identifier and the selected key value from the
6 index; and

7 means for releasing stabilization of any pages in the index prior to any
8 callback to the data manager.

1 30. The query processing system of claim 29 wherein if one of the set of access
2 processes is waiting for an exclusive latch for accessing the index, the stabilization means is
3 allowed to copy the data identifier and the selected key value from the index and release
4 stabilization of any pages in the index prior to any callback to the data manager.

1 31. A relational database management system comprising:

2 a data manager coupled to an index manager wherein the data manager including:

3 means for performing an index-data fetch wherein pushdown processing is

possible and means for requesting a callback from the index manager wherein the index manager comprises:

means for initiating a callback to the data manager for an index-specified row to determine if pushdown processing on the row is possible; and

means for continuing the index-data fetch by continuing to access an index and use the means for initiating a callback if the index manager receives a specific condition from a callback to the data manager.

32. The relational database management system of claim 31 wherein the data manager further comprises means for determining if pushdown processing is possible for a specified row and means for carrying out the processing and returning the specific condition.

33. The relational database management system of claim 32 wherein if pushdown processing is possible for a specified row, the data manager is allowed to carry out the processing and return the specific condition.

34. The relational database management system of claim 33 wherein the specific condition comprises a no data return condition.

35. The relational database management system of claim 34 wherein the index is stored on a plurality of pages and the index manager further comprises:

means for stabilizing a page containing a portion of the index to be accessed in the index to locate a data identifier for a selected key value wherein the means for stabilizing the

5 page does not release the stabilization of the page during a callback to the data manager.

1 36. The relational database management system of claim 34 wherein the index manager
2 further comprises:

3 stabilization means for determining if one of a set of access processes is
4 waiting for an exclusive latch for accessing, the stabilization means further comprising:

5 means for copying a data identifier and a selected key value from the index;
6 and

7 means for releasing stabilization of any pages in the index prior to any
8 callback to the data manager.

1 37. The relational database management system of claim 34 wherein if one of the set of
2 access processes is waiting for an exclusive latch for accessing the index, the stabilization
3 means is allowed to copy the data identifier and the selected key value from the index and
4 release stabilization of any pages in the index prior to any callback to the data manager.

1 38. A method for processing a database query in a database management system
2 comprising a data manager, a set of data, an index manager and an index on the data, the
3 data manager supporting index-data fetches, predicate checking and data consuming
4 operations, the method comprising the steps of:

- 5 a. calling the data manager to access query-specified data in the set of data,
6 b. for an index-data fetch, the data manager calling the index manager to locate
7 a data identifier in the index corresponding to a selected key value, and where a predicate

8 check or data consuming operation is possible, the data manager requesting a callback from
9 the index manager,

10 c. the index manager carrying out index manager index-data fetch steps by

11 i. locating a data identifier in the index for the selected key value and
12 where there is no data identifier for the selected key value, returning a result of no
13 index entry found,

14 ii. where a data identifier is located and a callback is requested, the index
15 manager initiating a callback to the data manager for determining whether the data
16 specified by the data identifier is to be returned to the runtime,

17 iii. the data manager locating the data specified by the data identifier and
18 determining whether a predicate check or consuming operation on the data are
19 possible and where such operations are possible carrying out the operations and
20 returning a no data return condition to the index manager, and

21 d. where the index manager receives a no data return condition from the
22 callback to the data manager, and where there is a next data identifier for the selected key
23 value in the index the index manager, the index manager continuing to carry out the index
24 manager index-data fetch steps for the next data identifier.

1 39. The method of claim 38, the index comprising pages having index nodes, the method
2 further comprising the steps of the index manager stabilizing the page containing the node to
3 be accessed in the index to locate the data identifier for the selected key value, and the index
4 manager not releasing the stabilization of the page during callback to the data manager.

1 40. The method of claim 38, the index comprising pages having index nodes, the
2 database management system supporting a set of access processes potentially able to access
3 the index, the method further comprising the step determining if a one of the set of access
4 processes is waiting for an exclusive latch for accessing the index, and if there is such an
5 access process, the index manager copying the data identifier and key value from the index
6 and releasing stabilization of any pages in the index prior to any callback to the data
7 manager.

1 41. A computer readable medium containing program instructions for processing a
2 database query in a database management system comprising a data manager, a set of data,
3 an index manager and an index on the data, the data manager supporting index-data fetches,
4 predicate checking and data consuming operations, the program instructions comprising the
5 steps of:

- 6 a. calling the data manager to access query-specified data in the set of data,
7 b. for an index-data fetch, the data manager calling the index manager to locate
8 a data identifier in the index corresponding to a selected key value, and where a predicate
9 check or data consuming operation is possible, the data manager requesting a callback from
10 the index manager,
11 c. the index manager carrying out index manager index-data fetch steps by
12 i. locating a data identifier in the index for the selected key value and
13 where there is no data identifier for the selected key value, returning a result of no
14 index entry found,
15 ii. where a data identifier is located and a callback is requested, the index

16 manager initiating a callback to the data manager for determining whether the data
17 specified by the data identifier is to be returned to the runtime,

18 iii. the data manager locating the data specified by the data identifier and
19 determining whether a predicate check or consuming operation on the data are
20 possible and where such operations are possible carrying out the operations and
21 returning a no data return condition to the index manager, and

22 d. where the index manager receives a no data return condition from the
23 callback to the data manager, and where there is a next data identifier for the selected key
24 value in the index the index manager, the index manager continuing to carry out the index
25 manager index-data fetch steps for the next data identifier.

1 42. The computer readable medium of claim 41, the index comprising pages having
2 index nodes, the method further comprising the steps of the index manager stabilizing the
3 page containing the node to be accessed in the index to locate the data identifier for the
4 selected key value, and the index manager not releasing the stabilization of the page during
5 callback to the data manager.

1 43. The computer readable medium of claim 41, the index comprising pages having
2 index nodes, the database management system supporting a set of access processes
3 potentially able to access the index, the method further comprising the step determining if a
4 one of the set of access processes is waiting for an exclusive latch for accessing the index,
5 and if there is such an access process, the index manager copying the data identifier and key
6 value from the index and releasing stabilization of any pages in the index prior to any

7 callback to the data manager.

8
1 44. A method for processing a database query in a database management system to return
2 data to a user, the database management system comprising a set of data, and an index on the
3 set of data, the database management system supporting an index-data fetch operation and
4 supporting a pushdown index-data fetch operation where a subset of the set of data is
5 accessed without the subset being returned to the user, the method comprising the steps of:

6 a. a data access step to locate an identified portion of the set of data
7 corresponding to a located data identifier where for an index-data fetch the data access step
8 uses an index access step to determine the located data identifier for a specified key value,

9 b. the index access step locating the data identifier in the index for the specified
10 key value, or returning a not found indicator where there is no data identifier in the index for
11 the specified key value,

12 c. where a database query is processed having a potential pushdown index-data
13 fetch, the index access step using a callback to perform the data access step for carrying out
14 the potential push-down process for successive data identifiers in the index for the specified
15 key value.

1 45. The method of claim 44, the index comprising pages having index nodes and in
2 which the index access step comprises the step of stabilizing the page containing the node to
3 be accessed in the index for locating the data identifier for the selected key value, and index
4 access step comprises the step of releasing the stabilization of the page only after completion

5 of any callback data access steps.

1 46. The method of claim 44, the index comprising pages having index nodes, the
2 database management system supporting a set of access processes potentially able to access
3 the index, the method further comprising the step determining if a one of the set of access
4 processes is waiting for an exclusive latch for accessing the index, and where a one of the
5 access processes is waiting, the index access step copying the data identifier and key value
6 from the index and releasing stabilization of any pages in the index prior to the callback to
7 perform the data access step.

1 47. A computer readable medium containing program instructions for processing a
2 database query in a database management system to return data to a user, the database
3 management system comprising a set of data, and an index on the set of data, the database
4 management system supporting an index-data fetch operation and supporting a pushdown
5 index-data fetch operation where a subset of the set of data is accessed without the subset
6 being returned to the user, the program instructions comprising the steps of:

7 a. a data access step to locate an identified portion of the set of data
8 corresponding to a located data identifier where for an index-data fetch the data access step
9 uses an index access step to determine the located data identifier for a specified key value,

10 b. the index access step locating the data identifier in the index for the specified
11 key value, or returning a not found indicator where there is no data identifier in the index for
12 the specified key value,

13 c. where a database query is processed having a potential pushdown index-data

14 fetch, the index access step using a callback to perform the data access step for carrying out
15 the potential push-down process for successive data identifiers in the index for the specified
16 key value.

1 48. The computer readable medium of claim 47, the index comprising pages having
2 index nodes and in which the index access step comprises the step of stabilizing the page
3 containing the node to be accessed in the index for locating the data identifier for the
4 selected key value, and index access step comprises the step of releasing the stabilization of
5 the page only after completion of any callback data access steps.

1 49. The computer readable medium of claim 47, the index comprising pages having
2 index nodes, the database management system supporting a set of access processes
3 potentially able to access the index, the method further comprising the step determining if a
4 one of the set of access processes is waiting for an exclusive latch for accessing the index,
5 and where a one of the access processes is waiting, the index access step copying the data
6 identifier and key value from the index and releasing stabilization of any pages in the index
7 prior to the callback to perform the data access step.

1 50. A query processing system comprising a data manager, a set of data, an index
2 manager and an index on the set of data, the data manager supporting index-data fetches, and
3 predicate checking and data consuming operations,

4 the data manager comprising:

5 a callback-enabling component for an index-data fetch where a predicate check or

6 data consuming operation is possible, the callback-enabling component requesting a callback
7 from the index manager,

8 a callback-execution component for determining if a predicate check or consuming
9 operation on a defined subset of the set of data is possible and where such operations are
10 possible, carrying out the operations and returning a no data return condition,

11 the index manager comprising

12 a data manager callback component for initiating a callback to the data manager for
13 an index-specified subset of the ordered set of data to determine if a predicate check or
14 consuming operation on the index-specified defined subset of the set of data is possible,

15 an index access component for continuing the index-data fetch steps by locating the
16 next one of the data identifiers for the selected key value or returning a result of no index
17 entry found, the index access component being invoked where the index manager receives a
18 no data return condition from a callback to the data manager.

1 51. The query processing system of claim 50, the index being stored on pages, the index
2 manager further comprising a stabilizing component for stabilizing the page containing a
3 portion of the index to be accessed to locate a data identifier for a selected key value, the
4 stabilizing component not releasing the stabilization of the page during callback to the data
5 manager.

1 52. The query processing system of claim 50, the index being stored on pages, the query
2 processing system supporting a set of access processes potentially able to access the index,
3 the query processing system further comprising a stabilization component for determining if

4 a one of the set of access processes is waiting for an exclusive latch for accessing the index,
5 and where there is such an access process, the stabilization component copying the data
6 identifier and key value from the index and releasing stabilization of any pages in the index
7 prior to any callback to the data manager.

1 53. In a relational database management system comprising table data and an index on
2 the table data, an improved data manager and an improved index manager,
3 the data manager supporting pushdown processing and comprising
4 a callback-enabling component for an index-data fetch where pushdown processing
5 is possible, the callback-enabling component requesting a callback from the index manager,
6 a callback-execution component for determining if for a specified row, pushdown
7 processing is possible and where such processing is possible, carrying out the processing and
8 returning a no data return condition,
9 the index manager comprising
10 a data manager callback component for initiating a callback to the data manager for
11 an index-specified row in the table data to determine if pushdown processing on the row is
12 possible,
13 an index access component for continuing the index-data fetch steps by continuing
14 to access the index and use the data manager callback component where the index manager
15 receives a no data return condition from a callback to the data manager.

1 54. The improved data manager and improved index manager of claim 53, the index
2 being stored on pages, the index manager further comprising a stabilizing component for

3 stabilizing a page containing a portion of the index to be accessed to locate a data identifier
4 for a selected key value, the stabilizing component not releasing the stabilization of the page
5 during callback to the data manager.

1 55. The improved data manager and improved index manager of claim 53, the index
2 being stored on pages, the relational database management system supporting a set of access
3 processes potentially able to access the index, the index manager further comprising
4 stabilization component for determining if a one of the set of access processes is waiting for
5 an exclusive latch for accessing the index, and where there is such an access process, the
6 stabilization component copying the data identifier and key value from the index and
7 releasing stabilization of any pages in the index prior to any callback to the data manager.